```
Start coding or generate with AI.
```

NUS Graded Assignment 3.1: ANN for Medical Diagnosis

Syed Salman Rabbani

Double-click (or enter) to edit

```
# Task 1: Utilise Libraries/Dataset

# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
from sklearn.utils.class_weight import compute_class_weight
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

from imblearn.over_sampling import SMOTE

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau


from google.colab import files
uploaded = files.upload()

import io
filename = list(uploaded.keys())[0]
df = pd.read_csv(io.BytesIO(uploaded[filename]))

df.head()
```

Choose Files enhanced_…_dataset.csv

- **enhanced_diabetes_dataset.csv**(text/csv) - 124845 bytes, last modified: 4/27/2025 - 100% done

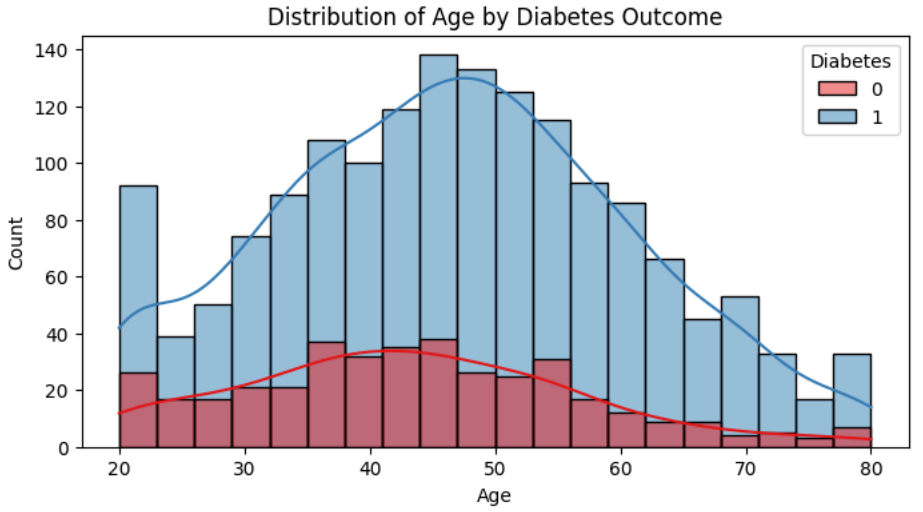
Saving enhanced_diabetes_dataset.csv to enhanced_diabetes_dataset.csv

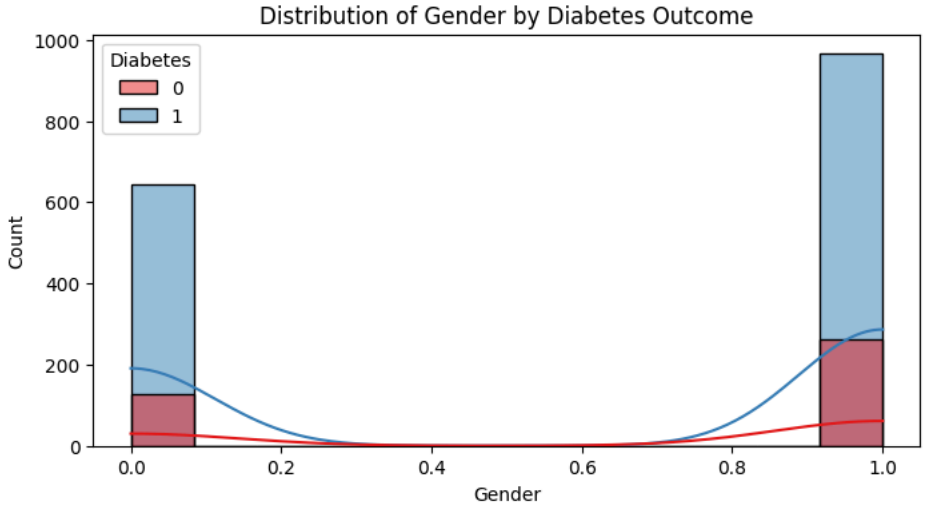
	Age	Gender	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	HbA1c	FastingBS	Trigly
0	52	1	1	124	95.0	20.0	0	36.0	0.078	4.7	87	
1	43	1	2	144	86.0	23.0	3	32.9	0.118	6.3	135	
2	55	1	4	141	113.0	18.0	297	43.4	0.139	5.4	113	
3	68	1	3	125	110.0	32.0	210	22.7	0.197	5.2	93	
4	41	0	0	177	92.0	14.0	189	38.1	0.078	10.5	126	

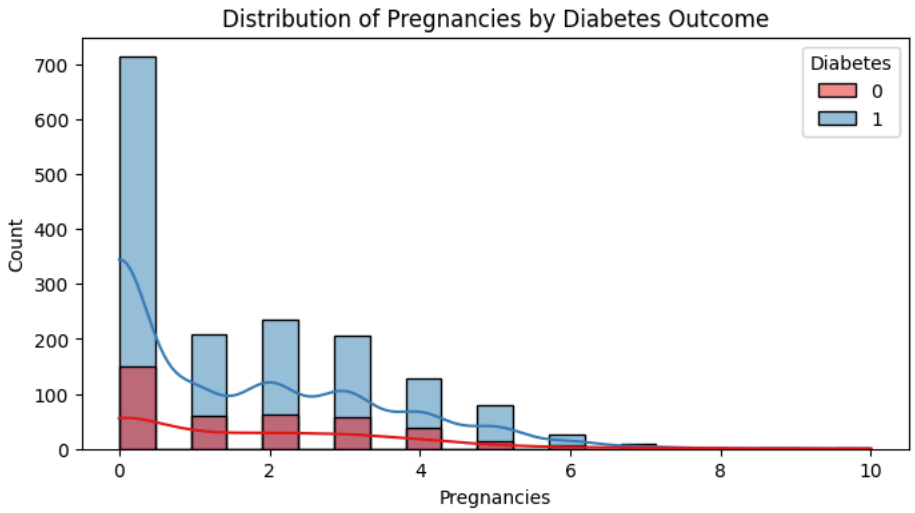
Next steps: Generate code with df | View recommended plots | New interactive sheet

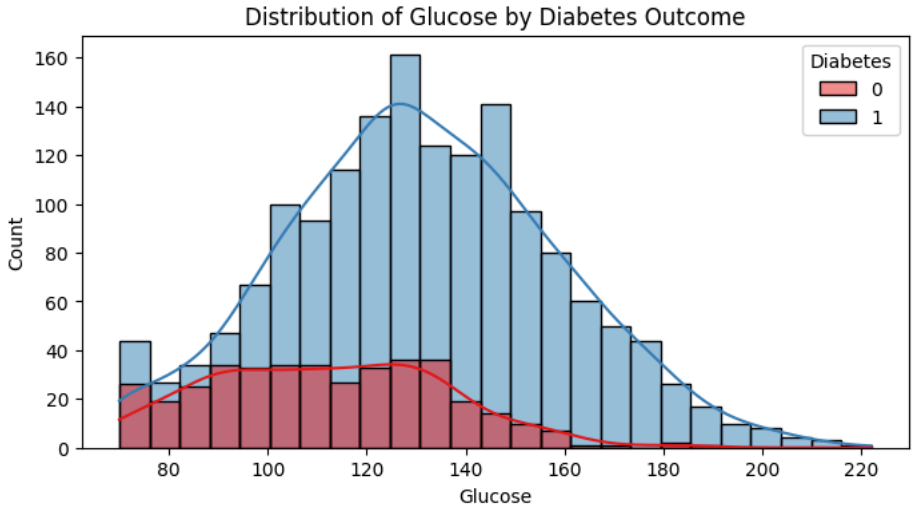
```
# Task 2: Generate at least three EDA visualisations

# Distribution plots comparing features between diabetic and non-diabetic patients
features = df.columns[:-1]
for feature in features:
    plt.figure(figsize=(8, 4))
    sns.histplot(data=df, x=feature, hue=df.columns[-1], kde=True, palette='Set1')
    plt.title(f'Distribution of {feature} by Diabetes Outcome')
    plt.show()

# For Correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Features')
plt.show()

# Feature importance , I am using Random Forest here
from sklearn.ensemble import RandomForestClassifier

X = df.drop(columns=[df.columns[-1]])
```

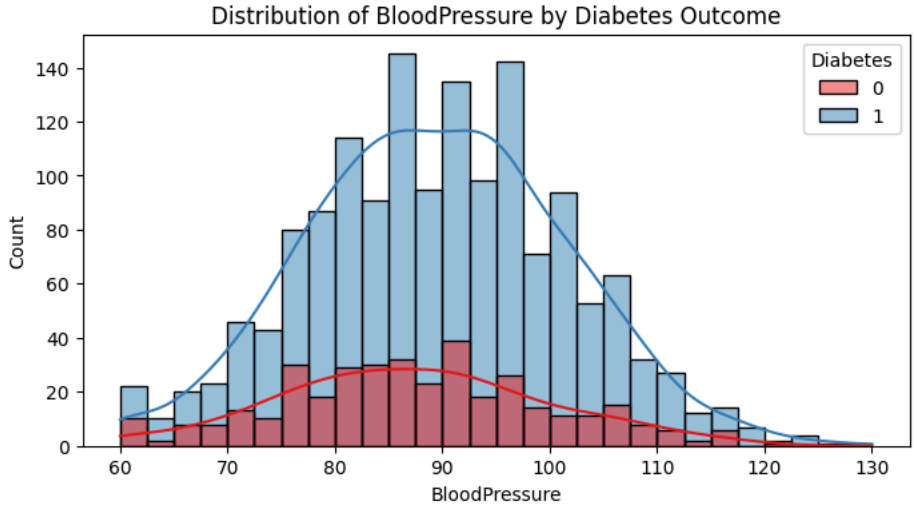
```
y = df[df.columns[-1]]

rf = RandomForestClassifier(random_state=42)
rf.fit(X, y)

importances = rf.feature_importances_
indices = np.argsort(importances)[::-1]
features_sorted = [features[i] for i in indices]

plt.figure(figsize=(10, 6))
sns.barplot(x=importances[indices], y=features_sorted)
plt.title('Feature Importance with Random Forest Model')
plt.show()
```

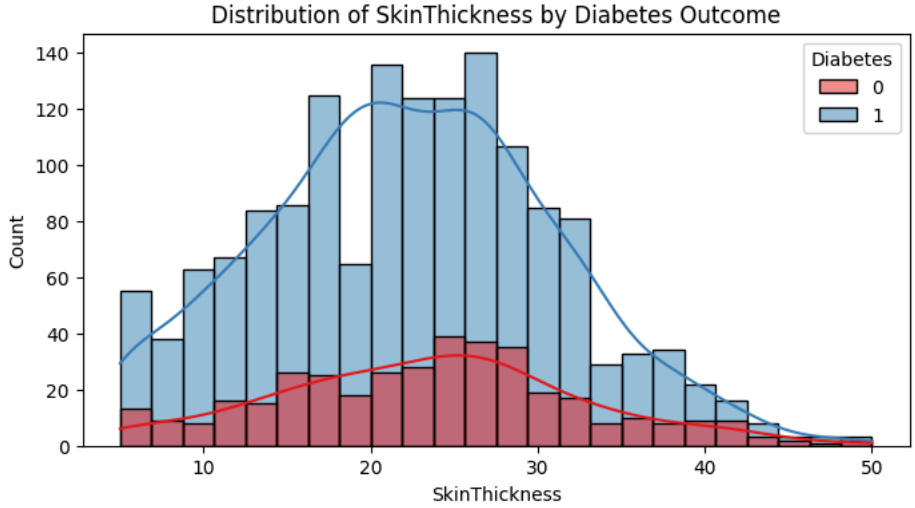

Distribution of Age by Diabetes Outcome
Diabetes
0
1
Count
Age

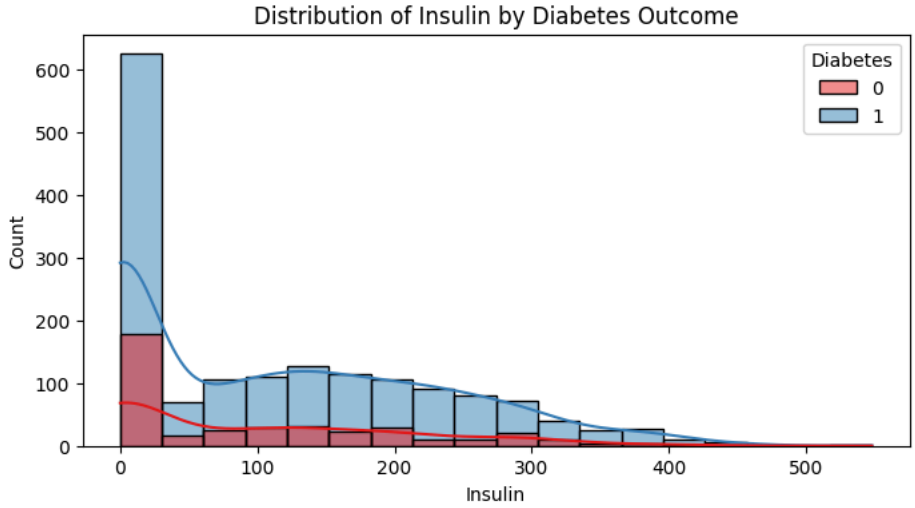

Distribution of Gender by Diabetes Outcome
Diabetes
0
1
Count
Gender

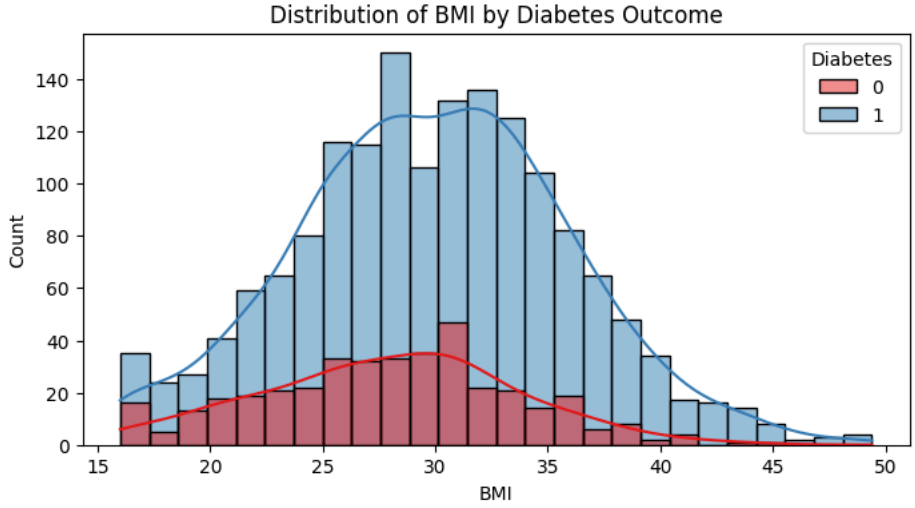

Distribution of Pregnancies by Diabetes Outcome
Diabetes
0
1
Count
Pregnancies

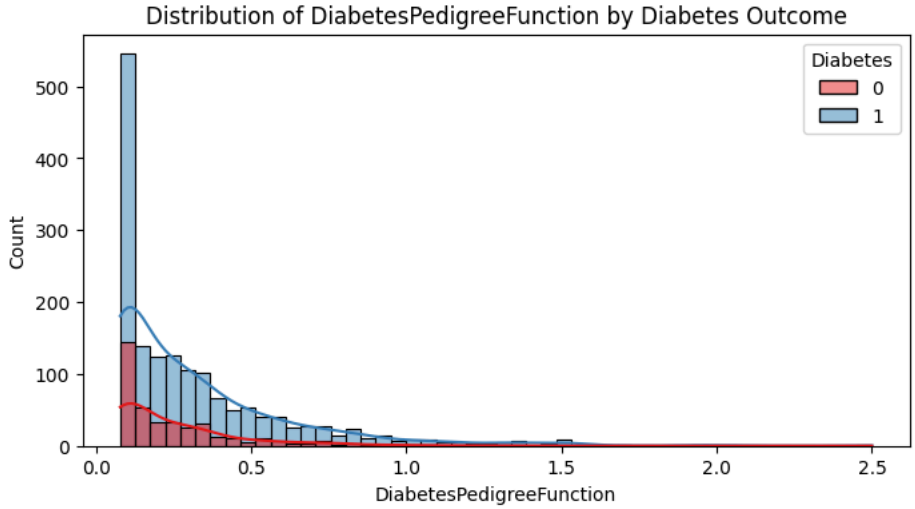

Distribution of Glucose by Diabetes Outcome
Diabetes
0
1
Count
Glucose

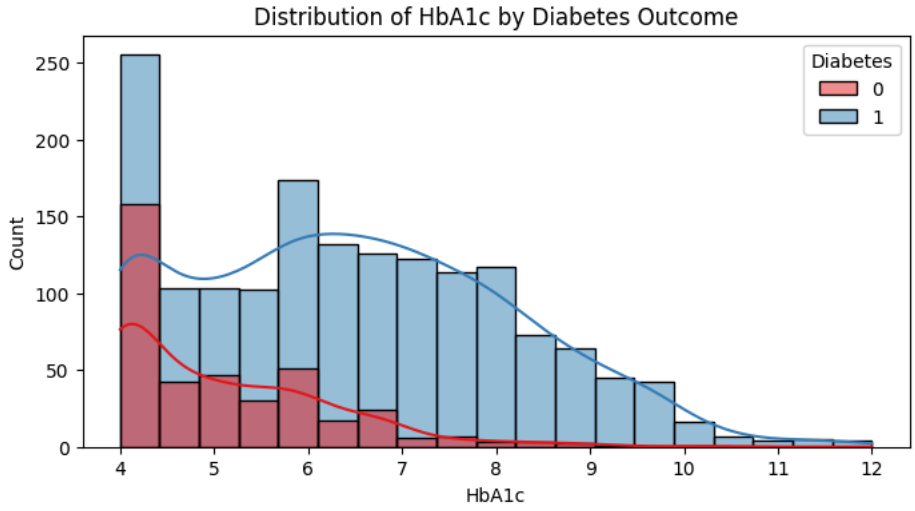

Distribution of BloodPressure by Diabetes Outcome
Diabetes
0
1
Count
BloodPressure

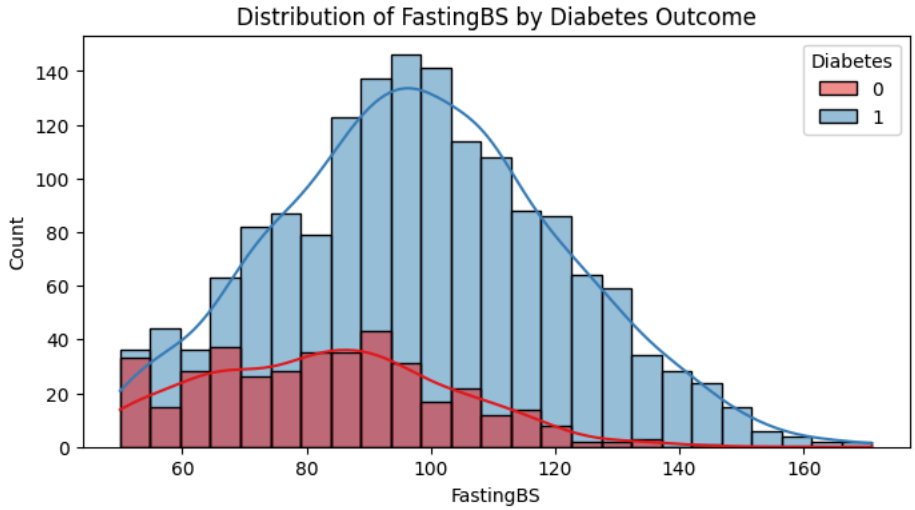

Distribution of SkinThickness by Diabetes Outcome
Diabetes
0
1
Count
SkinThickness

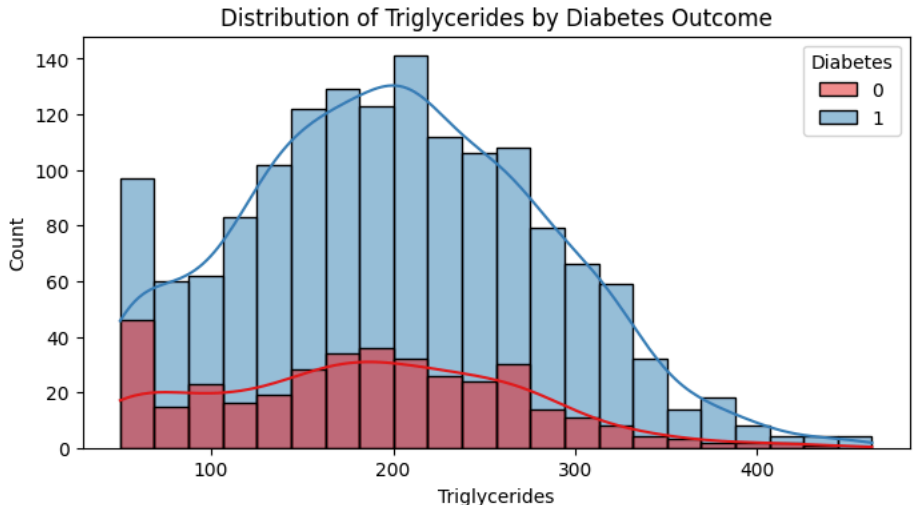

Distribution of Insulin by Diabetes Outcome
Diabetes
0
1
Count
Insulin

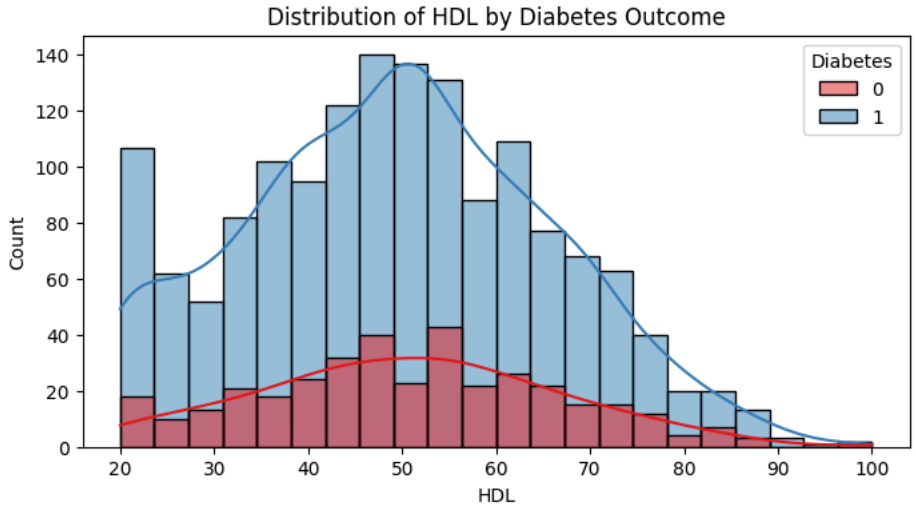

Distribution of BMI by Diabetes Outcome
Diabetes
0
1
Count
BMI

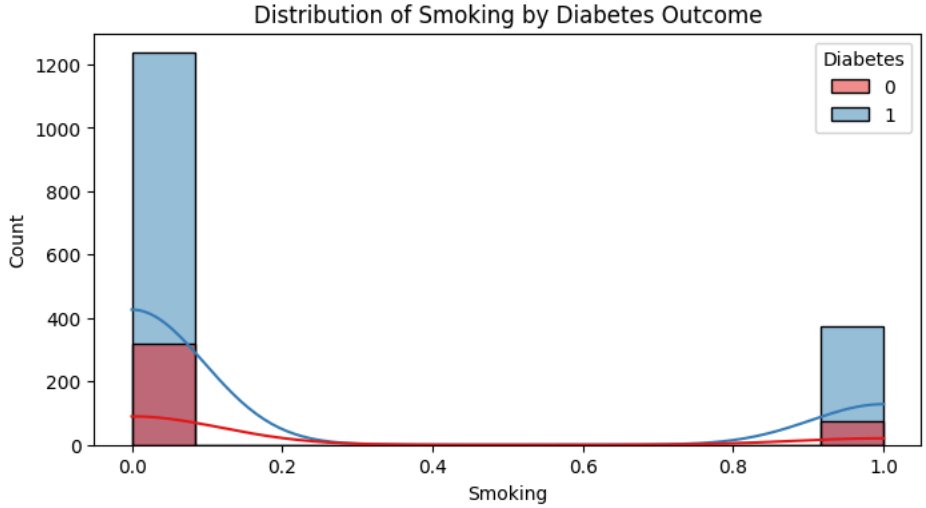

Distribution of DiabetesPedigreeFunction by Diabetes Outcome
Diabetes
0
1
Count
DiabetesPedigreeFunction

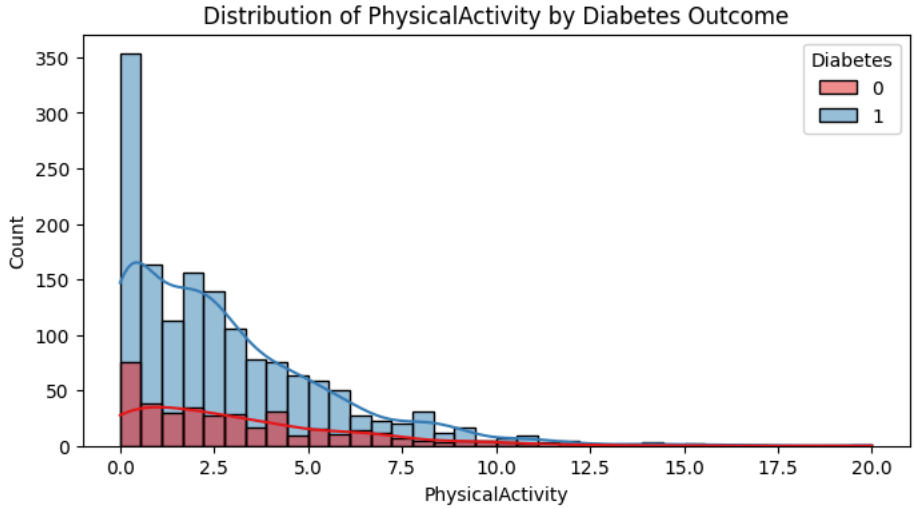

Distribution of HbA1c by Diabetes Outcome
Diabetes
0
1
Count
HbA1c

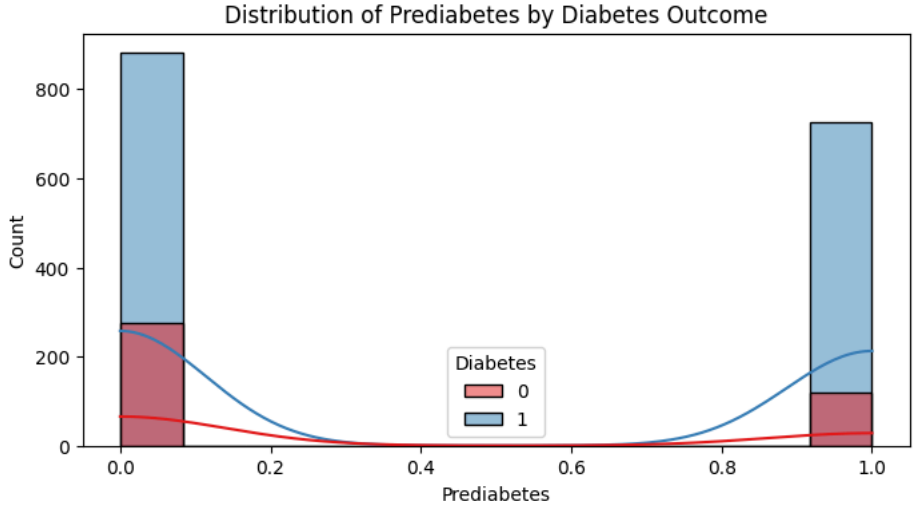

Distribution of FastingBS by Diabetes Outcome
Diabetes
0
1
Count
FastingBS

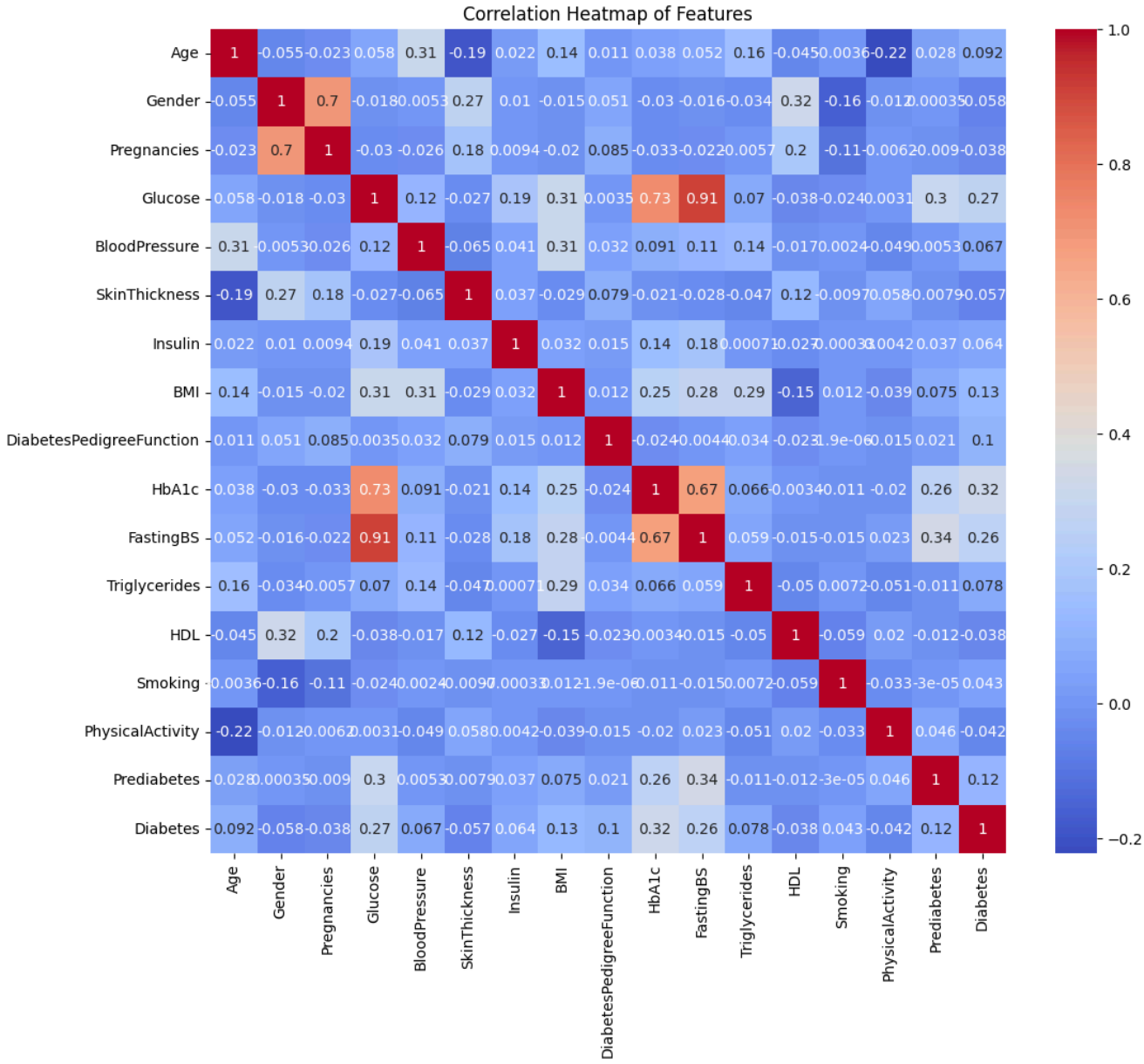

Distribution of Triglycerides by Diabetes Outcome
Diabetes
0
1
Count
Triglycerides

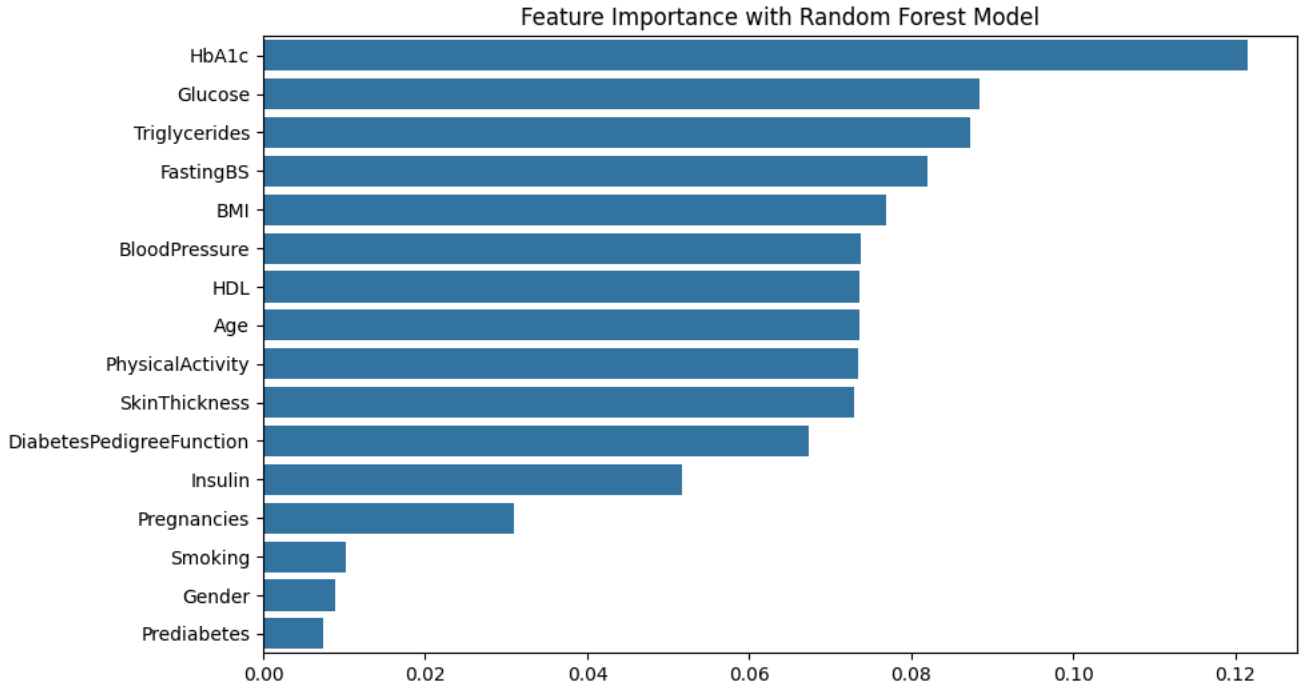

Distribution of HDL by Diabetes Outcome
Diabetes
0
1
Count
HDL


Distribution of Smoking by Diabetes Outcome
Diabetes
0
1
Count
Smoking


Distribution of PhysicalActivity by Diabetes Outcome
Diabetes
0
1
Count
PhysicalActivity

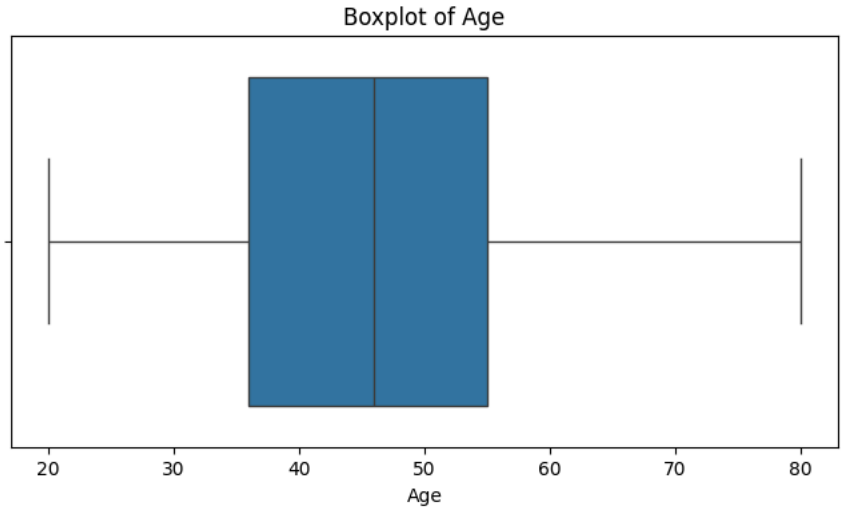

Distribution of Prediabetes by Diabetes Outcome
Diabetes
0
1
Count
Prediabetes

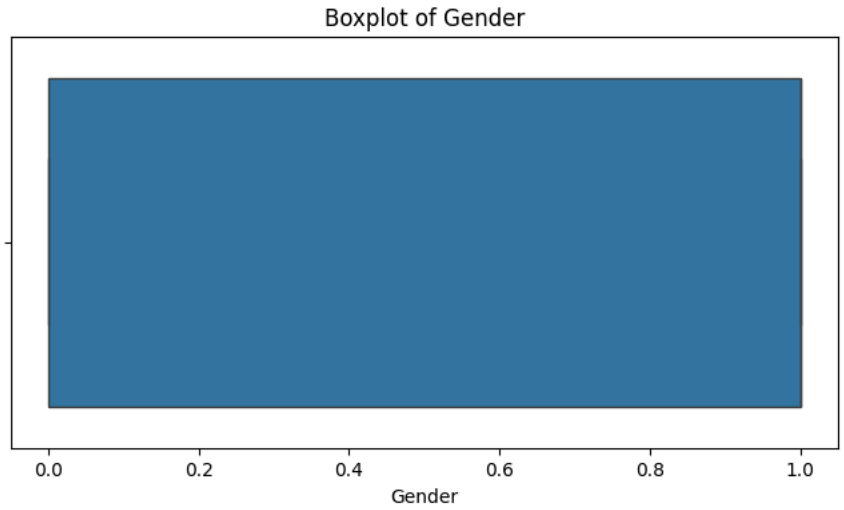

Correlation Heatmap of Features
Age
Gender
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
HbA1c
FastingBS
Triglycerides
HDL
Smoking
PhysicalActivity
Prediabetes
Diabetes
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

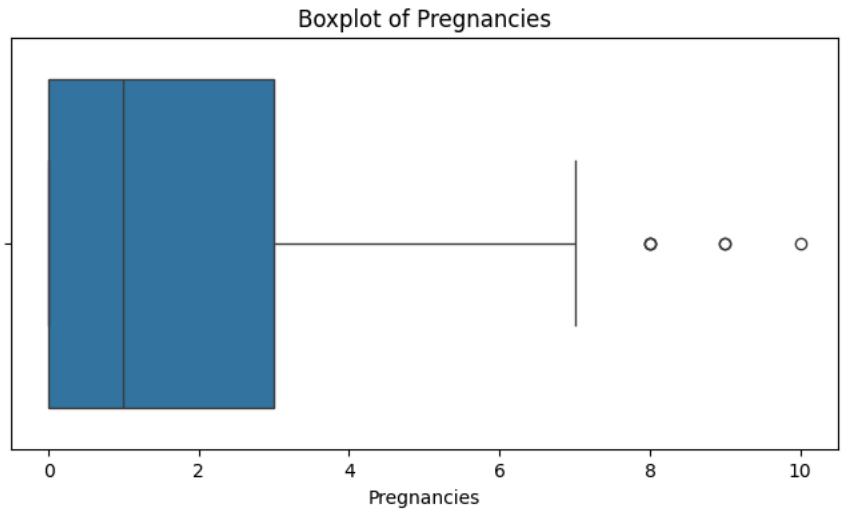

Feature Importance with Random Forest Model
HbA1c
Glucose
Triglycerides
FastingBS
BMI
BloodPressure
HDL
Age
PhysicalActivity
SkinThickness
DiabetesPedigreeFunction
Insulin
Pregnancies
Smoking
Gender
Prediabetes
0.00
0.02
0.04
0.06
0.08
0.10
0.12

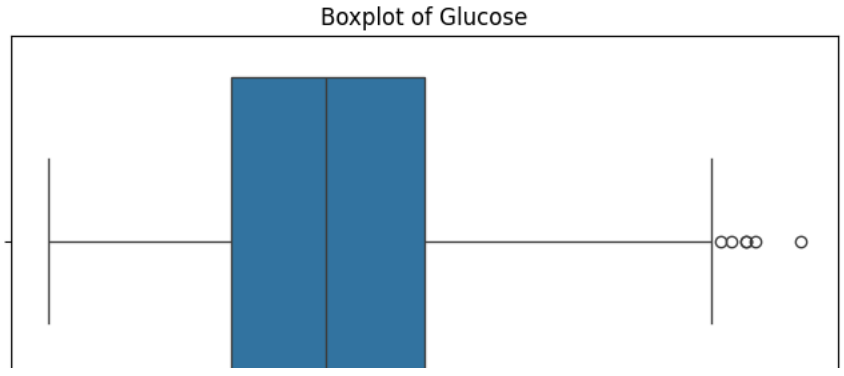
```
# Task 3: Analyse data quality

# To Check for missing or zero values
zero_features = ['SkinThickness', 'Insulin', 'BMI']
for feature in zero_features:
    missing_count = (df[feature] == 0).sum()
    print(f"{feature} has {missing_count} zero values.")

# Replace zero values with median to level up the data
for feature in zero_features:
    median = df[feature].median()
    df[feature] = df[feature].replace(0, median)

# Verifying if zeros are replaced
for feature in zero_features:
    missing_count = (df[feature] == 0).sum()
    print(f"{feature} after replacement has {missing_count} zero values.")

# Outlier detection using boxplots
for feature in features:
    plt.figure(figsize=(8, 4))
    sns.boxplot(x=df[feature])
    plt.title(f'Boxplot of {feature}')
    plt.show()
```

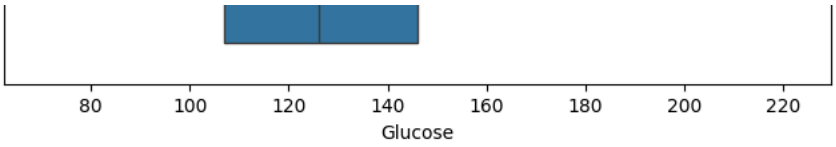
```
SkinThickness has 0 zero values.
Insulin has 737 zero values.
BMI has 0 zero values.
SkinThickness after replacement has 0 zero values.
Insulin after replacement has 0 zero values.
BMI after replacement has 0 zero values.
```

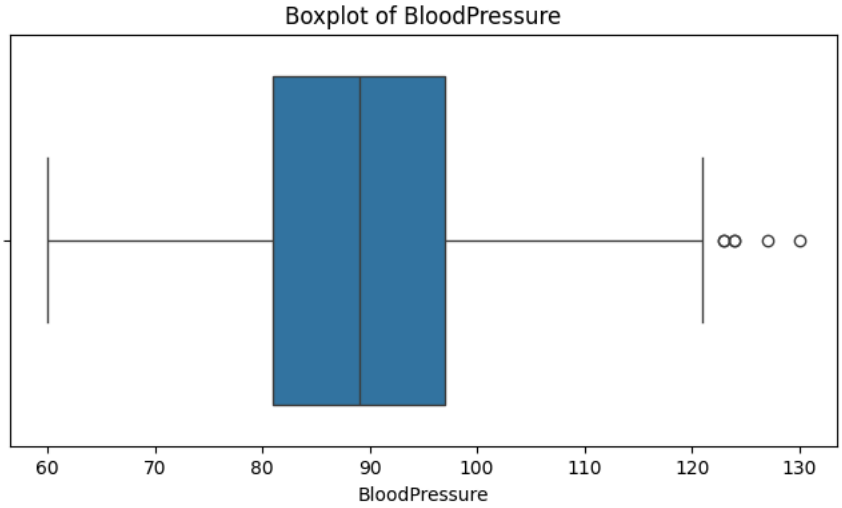


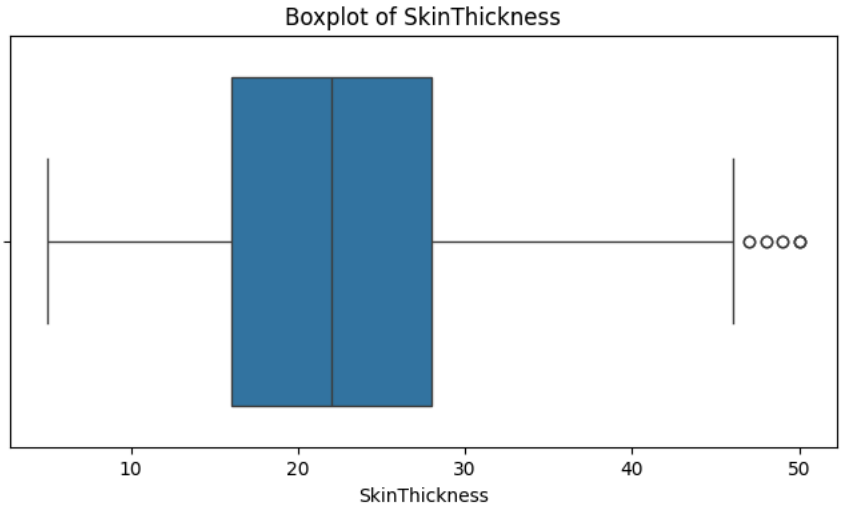


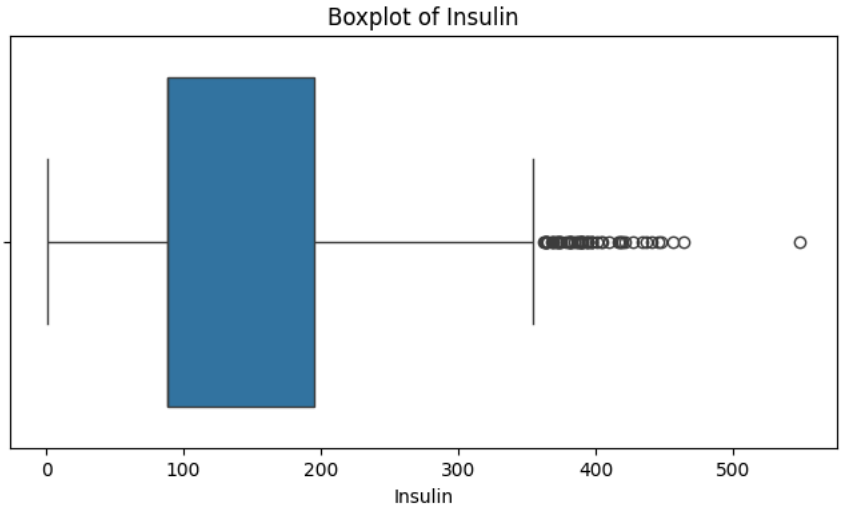


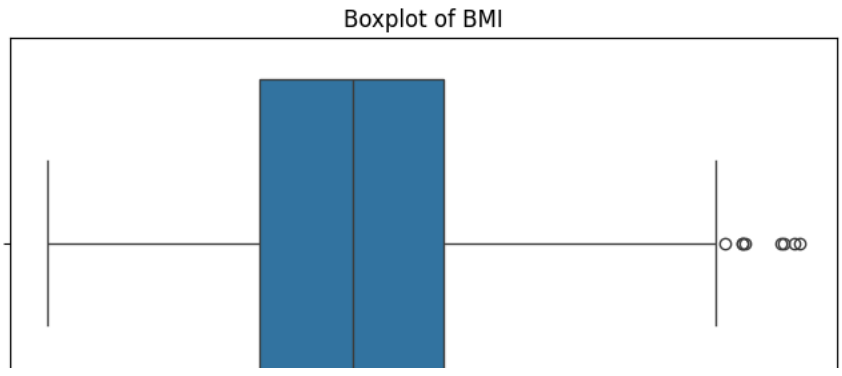



80
100
120
140
160
180
200
220
Glucose


Boxplot of BloodPressure
60
70
80
90
100
110
120
130
BloodPressure


Boxplot of SkinThickness
10
20
30
40
50
SkinThickness


Boxplot of Insulin
0
100
200
300
400
500
Insulin


Boxplot of BMI